Kanopian SDK Reference

# Overview

# REST API

## Overview

## MethodCall.php

#### Example

http://collect.idaxdata.com/MethodCall.php/**ClassName**::**MethodName**?*token*=&*deviceid*= &*compressresponse*=&*utf8encode*=&*app*=&*version*=&*build*=

#### Description

MethodCall.php is the entry point for all REST APIs. It accepts HTTP GET and POST methods. For POST methods, all GET parameters are ignored. Thus all parameters must be included in the body when using POST.

#### Required Parameters

* ClassName: Name of the class that implements the method
* MethodName: Name of the method that implements the REST API

#### Optional parameters

* token
  + Description: identifies the client. This parameter is required by most methods to identify the user and determine the user’s role. See AccountManager for information about users, roles, and acquiring a token.
  + Values: value acquired from the AccountManager::Login method.
  + Default value: empty string.
* deviceid
  + Description: Intended for usage when using the REST API from a device such as an iPhone or iPad.
  + Values: Unique identifier of the device.
  + Default value: empty string
* compressresponse
  + Description: This indicates if the response from the REST API should be compressed before sending it to the client. The client is responsible for decompressing the response using a ZLIB-compatible library.
  + Values: 0 – don’t compress, 1 - compress
  + Default value: 0
* utf8encode
  + Description: This indicates if the response from the REST API should be UTF-8 encoded before sending it to the client.
  + Values: 0 – don’t UTF-8 encode the response, 1 – UTF-8 encode the response.
  + Default value: 1
* app
  + Description: This indicates the name of the client application.
  + Values: Currently unused.
  + Default value: “”
* version
  + Description: This indicates the software version of the client application.
  + Values: Currently unused.
  + Default value: “”
* build
  + Description: This indicates the software build number of the client application.
  + Values: Currently unused.
  + Default value: “”

#### Response

Responses from MethodCall.php are JSON-encoded text files. The typical structure is

{

"results":

{

"response":"success",

"responder":"**ClassName**::**MethodName**",

"returnval":

{

"Method-specific response structure",

"resultstring":"information string"

}

}

}

“response” values

* “success” – the method call succeeded.
* “failed” – the method call failed.

“responder” value

* ClassName::MethodName – the class name and method as passed to MethodCall.php.

“returnval”

* Method-specific response structure
* “resultstring”: Most, but not all, methods will return a resultstring if the method fails. It can also be set for an method that succeeds. In that case, it has an informational or warning message. “resultstring” is intended to be used for debugging failures. In some cases, it has information that may be helpful for the user.

# Core

## Accounts

Every user in the system has an account. User accounts have the following properties:

* id: unique identifier of the user.
* email: the user’s email address.
* firstname: the user’s first name.
* lastname: the user’s last name.
* password: a secret set by the user and required for authenticating the user.
* role: a value assigned by the system administrator to indicate the user’s level of access to the system.
* rating: a value assigned by a project manager to indicate the user’s capability.
* state: an account can be in one of the following states:
  + REGISTERING: when the account is created by the administrator, email is sent to the user’s email address with a link the user can click to activate the account. The account remains in the REGISTERING state until the user has confirmed the email address by clicking the link in the email.
  + INACTIVE: the account is inactive. It must be activated by the administrator before any other actions can happen.
  + ACTIVE: the account is active and the user can authenticate with the system.
  + LOCKED: the account has been locked because too many successive authentication attempts failed. Any further attempt to authenticate, even with the correct password, will fail. The account must be unlocked by the administrator before authentication will be allowed again.
  + VALIDATING: the account email address has been changed, either by the user or by the administrator. The user must validate the email address by clicking the link sent in the email sent to notify the user of the change.
  + SETPASSWORD: The user has requested a “forgot password” link. The user must use the link to set a new password before being permitted to log in again. Once a new password has been set, the account state will become ACTIVE.

## Account Roles

Account roles define the user’s access to aspects / features of the system. The roles are hierarchical in that roles of higher levels inherit the access of the roles with lower levels.

These are the levels in increasing order.

* **User**: Can use basic system services. Counter.
* **QC**: Can do quality checks.
* **Designer**: Can create designs.
* **Project Manager**: Can administer projects.
* **Administrator**: Can administer user accounts.

## Tokens

A token is issued when a user is authenticated by providing a valid email and password combination. The token is then passed to calls that require an authenticated user to determine the user identity and role.

Method descriptions indicate the minimum role necessary to use that method. Those methods use the ‘token’ parameter passed to MethodCall.php to determine if the user has privilege to use the method. If not, access is denied and the method will fail.

For method that indicate “None” for the minimum role, it is not required to pass the ‘token’ parameter.

## Class AccountManager

### ActivateUserAccount

#### Description

Sets the account’s state to REGISTERING and sends the user a link that allows the user to confirm the account’s email address.

#### Minimum Role

Administrator

#### Parameters

* accountid: the id of the account to activate.

#### Response

Standard response

### ChangePassword

#### Description

Allows the user to change the account’s password.

#### Minimum Role

User

#### Parameters

* oldpassword: the account’s current password.
* newpassword: the new password.

#### Response

Standard response

### CreateAccount

#### Description

Creates a new account.

#### Minimum Role

Administrator

#### Parameters

* email: the email address for the new account. All account notifications will be sent to this email address.
* firstname: the first name of the new user.
* lastname: the last name of the new user.
* role: the role of the new user.

#### Response

“returnval“

{

accountid" – the identifier of the newly created account.

}

### DeactivateUserAccount

#### Description

Deactivates the user account by putting it in the “INACTIVE” state. Once deactivated, the user is no longer able to login to the system.

#### Minimum Role

Administrator

#### Parameters

* accountid

#### Response

Standard response

### ExecuteURL

#### Description

When there is a user-required action such as resetting a password, a URL is sent to the user in email. The URL contains a URL key that the server can use to determine information about the URL, such as the account the URL is intended for, its type and expiration date. When the user clicks the URL, it will invoke the ExecuteURL method.

#### Minimum Role

None

#### Parameters

* urlkey: identifier of the URL.

#### Response

“returnval“

{

“type" – the type of the URL.

“nextaction” – the next action required of the user.

}

“type” values:

* + “account\_registration”
  + “validate\_email”
  + “reset\_password”

“nextvalue” values:

* + “set\_password”: this is the next action required for “account\_registration” and “reset\_password” types, indicating the user must set a password on the account before it becomes active.
  + “none”: no other action required of the user.

### ExecuteURLSetPassword

#### Description

This method should be called after a call to ExecuteURL that specifies a “nextaction” of “set\_password”. On success, the password for the account will be set to the value provided in the “password” parameter.

#### Minimum Role

None

#### Parameters

* urlkey: identifier of the URL
* password: the password to set on the account.

#### Response

Standard response

### GetAccount

#### Description

Returns basic information for the user’s account.

#### Minimum Role

User

#### Parameters

None

#### Response

“returnval“

{

“email" – the email address for the account.

“firstname” – the user’s first name.

“lastname” – the user’s last name.

}

### GetUserAccount

#### Description

Returns basic information for the specified user’s account.

#### Minimum Role

User

#### Parameters

* accountid – the account identifier for the account for which to retrieve the information.

#### Response

“returnval“

{

“account”

{

“accountid” – the account identifier.

“email" – the email address for the account.

“firstname” – the user’s first name.

“lastname” – the user’s last name.

“role” – the user’s role.

“rating” – the user’s rating.

}

}

### GetUserAccounts

#### Description

Returns full information for all user accounts.

#### Minimum Role

Administrator

#### Parameters

None

#### Response

“returnval“

{

“accounts”

[

{

“accountid” – the account identifier.

“email" – the email address for the account.

“state” – the state of the account.

“creationtime” – time the account was created.

“registeredtime” – the time the account was registered.

“lastlogintime” – time of the user’s last login.

“firstname” – the user’s first name.

“lastname” – the user’s last name.

“role” – the user’s role.

“rating” – the user’s rating.

}

]

}

### Login

#### Description

The login method authenticates the user by verifying the email and password are a valid combination. Upon successful login, a token is generated that can be used in subsequent method calls made on behalf of the user.

#### Minimum Role

None

#### Parameters

* email: the email address of the account.
* password: the password of the account.

#### Response

“account”

{

“firstname” – the user’s first name.

“lastname” – the user’s last name.

“role” – the user’s role.

“rating” – the user’s rating.

“token” – the token to be used in subsequent method calls.

“tokenvalidityperiod” – the amount of time, in seconds, that the token is valid for.

}

### Logout

#### Description

Invalidates a user token so that it can no longer be used. The user must acquire a new token by logging in via AccountManager::Login.

#### Minimum Role

User

#### Parameters

None

#### Response

Standard response

### SendPasswordResetLink

#### Description

This will send a “reset password” URL to the user specified by the email parameter. This method is typically invoked when the user has forgotten the password and clicks on “forgot password”.

#### Minimum Role

None

#### Parameters

* email – the email address of the account.

#### Response

Standard response

### UpdateAccount

#### Description

#### Minimum Role

User

#### Parameters

* firstname: OPTIONAL - new first name to set on the account.
* lastname: OPTIONAL - new last name to set on the account.
* email: OPTIONAL – new email address to set on the account. Note that the account state will change to VALIDATING and the user will not be able to login again until the new email address has been validated.

#### Response

Standard response

### UpdateUserAccount

#### Description

#### Minimum Role

Administrator

#### Parameters

* accountid: identifier of the account to update.
* email: OPTIONAL – new email address to set on the account. Note that the account state will change to VALIDATING and the user will not be able to login again until the new email address has been validated.
* firstname: OPTIONAL - new first name to set on the account.
* lastname: OPTIONAL - new last name to set on the account.
* role: OPTIONAL – new role to set on the account.
* rating: OPTIONAL – new rating to set on the account.

#### Response

Standard response

## Class IssueManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

IDAX SDK Reference

# Overview

# Common

## Class JobManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

## Class JobSiteManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

# Tube

## Class ReportManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

# Video

## Class VideoJobSiteManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

## Class LayoutManager

### APIName

#### Description

#### Minimum Role

#### Parameters

#### Response

## Class KapturrKamManager

### QueryStatus

#### Description

#### Minimum Role

#### Parameters

* kamid

#### Response

#### Description

#### Minimum Role

#### Parameters

#### Response

### GetSessions

#### Description

#### Minimum Role

#### Parameters

* kamid

#### Response

### AddScheduledCaptureSession

#### Description

#### Minimum Role

#### Parameters

* kamid
* starttime
* endtime

#### Response

### UpdateScheduledCaptureSession

#### Description

#### Minimum Role

#### Parameters

* kamid
* sessionid,
* starttime
* endtime

#### Response

### DeleteSessions

#### Description

#### Minimum Role

#### Parameters

* kamid
* sessionid\_{$i}

#### Response

### UploadVideos

#### Description

#### Minimum Role

#### Parameters

* kamid
* sessionid\_{$i}

#### Response

### ControlLiveStreaming

#### Description

#### Minimum Role

#### Parameters

* kamid
* startstream

#### Response